(19) Europäisches Patentamt
European Patent Office
Office européen des brevets

(11) Publication number: **0 239 359**
A2

(12) EUROPEAN PATENT APPLICATION

(21) Application number: **87302513.4**

(22) Date of filing: **24.03.87**

(51) Int. Cl.[4]: **G 06 F 12/10**

(30) Priority: **25.03.86 JP 64976/86**

(43) Date of publication of application: **30.09.87**
Bulletin 87/40

(84) Designated Contracting States: **CH DE FR GB IT LI NL SE**

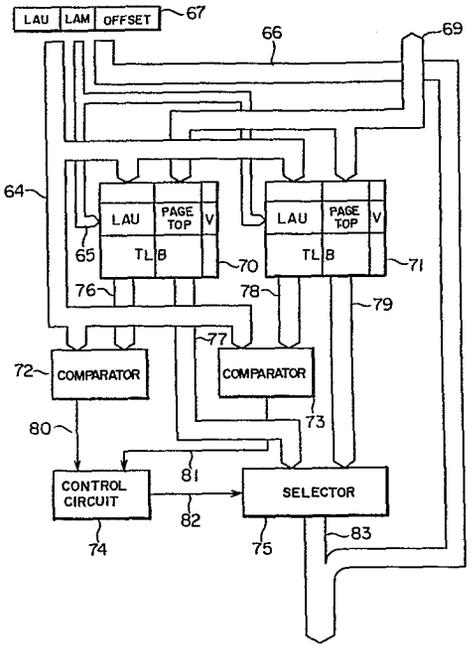
(71) Applicant: **HITACHI, LTD., 6, Kanda Surugadai 4-chome, Chiyoda-ku Tokyo 101 (JP)**

(72) Inventor: **Ozawa, Koji Kosakiyama Dai-9 Apartment 202, 6-1, Suwacho-4-chome, Hitachi-shi (JP)**
Inventor: **Araoka, Manabu, Tenjin Apartment A-202 1077-6, Ishinazakacho, Hitachi-shi (JP)**
Inventor: **Takaya, Soichi, Chishinryo 503 33-2, Nishinarusawacho-1-chome, Hitachi-shi (JP)**

(74) Representative: **Calderbank, Thomas Roger et al, MEWBURN ELLIS & CO. 2/3 Cursitor Street, London EC4A 1BQ (GB)**

(54) Address translation circuit.

(57) An address translation circuit for translating a logical address into a physical address in a computer system using a virtual storage method includes two high-speed buffers (TLB's) for an instruction and an operand, respectively. One of the buffer is selected for use at the time a memory access depending on a signal (6) supplied from a processing unit (1) to indicate whether the memory access is related to an instruction cycle or an operand cycle. This configuration enables a high-speed address translation without lowering the TLB hit rate and without increasing the amount of the hardware components.



EP 0 239 359 A2

ADDRESS TRANSLATION CIRCUIT

The present invention relates to a circuit for translating an address in a computer system using a virtual storage method.

A processor to support a virtual storage requires an address translation circuit for translating a logical address into a physical address. As such an address translation method, a segmentation/paging composite method is applied to a case where a large logical space is used. In this method, for example, the logical space is divided into several segments and each segment is further subdivided into a plurality of pages. There are disposed a segment table ST and a page table PT (these tables are referred to as translation tables) to indicate whether a particular page in a particular segment exists in the main storage or not and when they are found to be present, indicate their physical addresses of the main storage. Thus, the address translation in this method is achieved through a 2-stage translation by use of these two tables. This provision is adopted because it will increase the amount of the hardware elements to provide a table which establishes a one-to-one correspondence between all logical addresses in the logical space and the physical addresses.

Referring now to Fig. 1 which is an explanatory

diagram of the segmentation/paging composite method, the segmentation/paging composite method will be explained.

The segment table ST and the page table PT are provided in the main storage. A segment table origin register STOR provided in the address translation mechanism is used to store the top address STTOP of the segment table ST. By accessing an address in the segment table assigned or specified by the top address STTOP and the SEG field of a specified logical address LA, that top address PTTOP of the page table PT which corresponds to the segment (SEG) field of the logical address LA is read and then the obtained top address PTTOP is set in a page table register (not shown) provided in the address translation mechanism. Subsequently, by accessing an address in the page table PT assigned or specified by the content of the page table register and the page field PAGE of the logical address, the top address PAGETOP of the page corresponding to the page field PAGE of the logical address LA is read. This read PAGETOP is the upper-order portion of the physical address. The lower-order portion OFFSET of the logical address LA is the lower-order portion of the physical address as it is. Thus, the physical address can be obtained by combining the PAGETOP and lower-order portion OFFSET of the logical address LA. However, in this address translation method, the tables ST and PT are accessed to translate a logical address into a physical address, namely, the main storage is twice accessed, which results in a time-consuming address

translation.

To overcome this, there can be considered an address translation method using a high-speed translation buffer or translation lookaside buffer (TLB) which functions like the translation table.

Fig. 2 is an explanatory diagram of an address translation method using such a TLB, and Fig. 3 shows an example of the logical address in this case. Namely, the logical address 67 of this example includes an upper-order portion LAU (17 bits), an intermediate portion LAM (3 bits), and a lower-order portion OFFSET (12 bits). Although the TLB 60 comprises eight entries corresponding to 3-bit LAM, only one entry is shown in the TLB 60 of Fig. 2. In each entry are registered the upper-order LAU of the logical address and the PAGETOP respectively corresponding to the top addresses of the segment and the page for an instruction or operand which was recently accessed and exists in the main storage. When the main storage is to be accessed, in order to determine whether or not the inputted logical address 67 exists in the TLB, the TLB is read to obtain an entry corresponding to the LAM of the logical address 67. The LAU 62 in the entry is then compared with the LAU 64 of the accessed logical address 67 by means of a comparator 61 whether they are identical (to be referred to as a TLB hit hereafter) or not identical (to be referred to as a TLB mishit). If a hit results, the PAGETOP in the entry and the OFFSET 66 of the accessed logical address 67 are combined

to generate a physical address 68. In the case of TLB mishit, since this PAGETOP is invalid, the address translation is accomplished according to the address translation method of Fig. 1 to obtain the LAU and the PAGETOP, which in turn are registered to an entry of the TLB. Consequently, TLB hit results for the subsequent access to the pertinent address. In this method, the main storage is not accessed in the case of TLB hit, and hence the address translation can be performed at a higher speed as compared with the method of Fig. 1.

Fig. 4 shows a method using one TLB described above. This method has been proposed in the JP-A-61-217846 laid-open on September 27, 1986 (Japanese Patent Application No. 60-57576 filed on March 23, 1985 in the name of the same assignee of the present application). The operation of this translation method is the same as the operation described in conjunction with Fig. 2. But, it is to be noted here that, in Fig. 4, the LAU 64 of the logical address 67 and the data bus 69 are inputted to the TLB. This is because the LAU and the PAGETOP corresponding to the data read from the main storage are to be registered to the TLB 60 when TLB mishit occurs.

Fig. 5 shows an example using two TLB's similar to that described in the JP-B-1658 and the like. Although the method for selecting an entry of the TLB is the same as in the case of one TLB scheme, since two TLB's are used in this method, two selectors 72 and 73 are necessary for the judgment of TLB hit by comparing the

LAU's 76 and 78 respectively registered to the TLB's 70 and 71 with the LAU 64 of the logical address 67 to be accessed. Further, there must be also provided a selector 75 for determining, based on the comparison results 80 and 81, which one of TLB's is indicating the TLB hit and selecting the PAGETOP 77 or 79 registered in the TLB which shows the TLB hit and a control circuit 74 for outputting a select signal 82 to control the selector 75. If hit does not occur in the TLB 70 nor in the TLB 71, the translation method of Fig. 1 is also executed and after the translation, for the registration of the entry to the TLB, LRU control method or the like is used, namely, an entry to which data has been more previously registered and which contains the old contents (LAU, PAGETOP, etc.) is updated with new contents.

The above-mentioned schemes using one or two TLB's have the following problems. Assume in the 1-TLB method of Fig. 4 that the logical address is represented by 8 hexadecimal characters and that the 5th character from the most significant bit thereof is an intermediate section LAM used to select an entry of the TLB as shown in Fig. 6. If a logical address at which an instruction is stored in the main storage is 3E503000, the entry number of the TLB is 3. Further, assuming that data to be used by the instruction is stored at a logical address 6F253000, the entry number of the TLB is also 3, which means that the instruction area and the data area use the same entry of the TLB. In such a case, when the

instruction cycle and the operand cycle are alternately executed, mishit occurs repetitiously, which leads to a problem that the hit rate is abruptly lowered.

On the other hand, in the 2-TLB method of Fig. 5, the situation associated with the problem of the 1-TLB method does not cause any problem bacause of two TLB's. In this method however, until the judgment of the TLB hit is finished for both TLB's, selection of one PAGETOP to be used as a part of the physical address out of two PAGETOP's registered in the TBL's must be deffered. Consequently, the speed of the address translation to obtain the physical address is lowered when compared with the 1-TLB method. Moreover, since a comparator is necessary for each TLB, there arises a problem that the amount of hardware elements is increased as compared with the 1-TLB method.

It is therefore an object of the present invention to provide an address translation circuit for use in a computer system using a virtual storage method, in which a high-speed address translation can be achieved without lowering the TLB hit rate and without greatly increasing the amount of the hardware components.

According to one aspect of the present invention, two TLB's are provided in a configuration in which one of the TLB's is used for an instruction and the other thereof is dedicated to an

operand and during the memory cycle, a signal is outputted from the processor to indicate whether the current cycle is an instruction cycle or an operand cycle, thereby determining one of the TLB's to be used.

More specifically, according to the present invention, there are provided a TLB for an instruction and a TLB for an operand so as to discriminately store data representing the position where the instruction is stored and data representing the position where the data or operand is stored, respectively. Consequently, even when the instruction and the data are linked to the same entry, the contents currently stored in the entry need not to be removed and hence mishits are not increased. Furthermore, based on a signal outputted from the processor to indicate whether the access is the instruction cycle or the operand cycle, one of the TLB's to be used is determined without waiting for the result of the TLB hit judgment, which enables the address translation to be conducted at as high a speed as the 1-TLB method.

The present inventin will be apparent from the following detailed description taken in conjunction with the accompanying drawings in which:

Fig. 1 is an explanatory diagram of a translation method to translate a logical address into a physical address;

Fig. 2 is a schematic diagram for explaining

the operaiton of a TLB;

Fig. 3 is a schematic diagram illustrating a configuration of a logical address when a TLB is used;

Fig. 4 is a configuration diagram of a translation circuit when one TLB is used;

Fig. 5 is a configuration diagram of a conventional translation circuit when two TLB's are used;

Fig. 6 is a schematic diagram useful for explaining the problem of a 1-TLB method;

Fig. 7 is a configuration diagram of a computer system using a virtual storage method to which the present invention is applicable;

Fig. 8 is a schematic block diagram showing a structure of the memory control unit of Fig. 7;

Fig. 9 is a block diagram of an embodiment of the address translation section of Fig. 8 according to the present invention; and

Fig. 10 is a simplified circuit diagram illustrating an embodiment of an address translation circuit according to the present invention.

Embodiments of the present invention will be described with reference to the accompanying drawings. In the drawings, the same reference numerals are assigned to the same members and the description thereof will not be repeatedly given unless such description is necessary.

An embodiment will be described according to

the present invention. Fig. 7 is a schematic diagram showing the overall configuration of a system to implement the present invention. This system comprises a basic processing unit (BPU) 1, a memory control unit (MCU) 2, a main storage (MS) 3, a file control processor (FCP) 4, and a disk unit (DISK) 5, which are connected by use of address lines 7-9, data lines 10-12, and control lines 13-15. A signal 6 indicates whether the memory access is related to an instruction cycle or an operand cycle. The file control processor 4 and the disk unit 5 are connected to each other by use of a data line 16, a function line 17, and a status line 18. The programs to be executed by the basic processing unit 1 are stored in the disk unit 5 and a portion thereof is loaded in the main storage 3 for execution.

Fig. 8 shows a configuration of the memory control processor 2 comprising an address translation section (MMU) 20, cache memory (CACHE) 22 for holding some of data stored in the MS 3, and a control circuit (MCUCTL) 21 for controlling the entire system. These constituent components are connected by use of internal signal buses 26-28 respectively corresponding to the address, data, and control; moreover, the signals on the respective signal buses are connected via the interface circuits 23-25 to the basic processing unit 1, the file control processor 4, and the main storage 3, respectively. In the case of a memory access from the basic processing unit 1, the address translation section

20 is supplied with a logical address via the address line 7, a signal 6 indicating whether the memory cycle is the instruction cycle or the operand cycle, and a control signal outputted from the control circuit 21. Here, the logical address is translated into a physical address, and then the cache memory 22 is accessed via the internal signal bus. And, when, as a result, the fact that data corresponding to the physical address is stored in the cache memory 22 is indicated and also the TLB hit judgment signal 36 outputted from the address translation section 20 indicates a hit state, the control circuit 21 notifies the end of access to the basic processing unit 1 via the interface circuit 23 and the interface 13. On receiving this notification of the end of access, the basis processing unit 1 fetches data from the cache memory 22 through the bus 27. If a mishit occurs in the cache memory 22, the main storage is accessed via the internal signal buses 26 and 28 and the interface circuit 25 and the data fetched from the main storage is transferred to the basic processing unit 1 and is written at the same time in the cache memory 22. In addition, accessing of the file control processor 4 by the basic processing unit 1 is achieved via the internal signal buses 26-28 and the interface circuit 24.

Fig. 9 shows a configuration diagram of the address translation section 20. A TLB 32 is a buffer for implementing a high-speed address translation, while a segment table origin register 33 and a page table register

34 are registers to be used for an address translation when mishit occurs for the TLB 32. When an access from the basic processing unit 1 is to be effected in the V=R space, the logical address is identical to the physical address; consequently, a selector 35 outputs as a physical address the logical address 19 supplied to the input A of the selector 35. When the access from the basic processing unit 1 is to be conducted in other than the V=R space and hit occurs for the TLB 32, the page top address registered in the entry of the TLB 32 for which hit has occurred and the offset field of the logical address 19 are combined to generate a physical address 37, which is then outputted to the input B of the selector 35. If mishit occurs for the TLB 32, the segment table on the main storage 3 is first referred to to obtain the top address of the page table. In this case, a signal 38 obtained by combining the content of the segment table origin register 33 with the segment field of the logical address 19 is outputted to the input C of the selector 35. Subsequently, the page table in the main storage 3 is referred to to obtain the page top address. In this case, a signal 39 obtained by combining the content of the page table register 34 with the page field of the logical address 19 is outputted to the input D of the selector 39. The operation of the selector 35 is controlled by the control signal 30 from the control circuit 21 of Fig. 9. Incidentally, the page table register 34 is beforehand set with the top address of the

page table outputted from the segment table through an access by the signal 38 supplied to the input C. And, when the signal 39 fed to the D input causes to access an address associated with this signal 39 in the page table in the main storage 3, the top address of the page corresponding to the page field of the logical address 19 is read. The obtained top address is the upper-order portion of the physical address. Since the lower-order portion of the physical address is identical to the lower-order portion OFFSET of the logical address 19, the top address and the OFFSET are conbined to generate the physical address. This is the same as in the case of Fig. 1. In this connection, in the case of the configurations of Figs. 9 and 8 and in the case of TLB hit, the logical address takes the form such that it is composed of the upper-order portion LAU, the intermediate portion LAM, and the lower-order portion OFFSET, as shown in Fig. 3, whereas in the case of TLB mishit, it takes the form such that it comprises the upper portion SEG and the intermediate portion PAGE as shown in Fig. 1 and the lower-order portion OFFSET which is the same as that of Fig. 3, as if there are two different logical addresses. However, these logical addresses are not two different ones but the same one, only the position dividing the upper-order portion and the intermediate portion being different. For example, assume that the logical address comprises 32 bits. In the case of Fig. 3, the OFFSET, LAM, and LAU are constituted from bits 0-11, 12-14

(3 bits), and 15-31 (17 bits), respectively; whereas, in the case of Fig. 1, the OFFSET, PAGE, and SEG portions are constituted from bits 0-11, 12-21 (10 bits, namely, 2^{10}=1024 pages), and 22-31 (10 bits), respectively.

The address translation circuit of the present invention resides in the TLB 32 of Fig. 9 and Fig. 10 is an embodiment of the present invention. The logical address 19 shown in Fig. 10 includes an upper-order LAU, an intermediate LAM (TLB entry selection bit), and a lower-order OFFSET as described in conjunction with Fig. 3. That is, the intermediate LAM is constituted of 3 bits and hence there are included 16 entries in total, namely, 8 entries each for the instruction TLB (ITLB) 40 and operand TLB (OTLB). When a memory access from the basic processing unit 1 is initiated, based on the intermediate portion 51 of the logical address 19 transferred via the address line 7 and the interface circuit 23, an entry is selected from the 8 entries each for the ITLB 40 and OTLB 41; and at the same time, a signal 6 supplied from the basic processing unit 1 to indicate whether or not the memory access is associated with the instruction cycle or the operand cycle is inputted to the selectors 42, 43 together with the logical address 19. For the instruction cycle, the upper-order portion 45 and page top address 47 of the ITLB 40 is selected by and outputted from the selectors 42 and 43. For the operand cycle, the upper-order portion 46 and page top address 48 of the OTLB 41 are selected by and outputted from the selectors 42, 43.

Since the selectors 42, 43 are in a select enable state before the entries are read from the TLB's 40, 41, the output signals 49 and 53 of the selectors are established as soon as the TLB output signals are established, which unnecessitates the time to be used for the selection.

The output signal 49 from the selector 42, namely, the selected upper-order portion of the address is compared with the upper-order portion of the logical address 19 by means of the comparator 44 to judge whether or not TLB hit occurs. On the other hand, the output signal 53 from the selector 43, namely, the selected page top address is conbined with the offset 52 of the accessing logical address 19 to generate and output a physical address 37, which enables the accessing of the cache memory 22 (Fig. 8) without waiting for the TLB hit judgment. The judge signal 36 outputted from the comparator 44 is fed to the control circuit 21. If hit results, the control circuit 21 indicates the physical address 37 to be valid and continues the access control on the cache memory 22 to effect a hit judgment to determine whether data corresponding to the physical address has been stored in the cache memory 22. If the hit judgment on the TLB 32 results in mishit, the physical address 37 is invalid and hence the control circuit 21 activates again the address translation section 20 by use of the control signal 30, which causes the operations to be execute to refer to the segment table and the page table by use of the registers 33, 34. The page top address 29 is

then registered in the TLB 32 together with the upper-order portion 50 of the logical address 19 by the control circuit 21.

According to the present invention, even when an instruction area and a data area to be used by the instruction are linked to the same entry of a TLB, the TLB hit rate is not deteriorated and a physical address can be outputted without waiting for the TLB hit judgment, which leads to an advantageous effect that the address translation can be conducted at a high speed.

While the present invention has been described with reference to the particular illustrative embodiments, it is not to be restricted by those embodiments but only by the appended claims. It is to be appreciated that those skilled in the art can change or modify the embodiments without departing from the scope and spirit of the presnet invention.

CLAIMS:

1. An address translation circuit for use in a virtual storage system in which a logical address inputted from processing means (1) at a time of memory access is translated into a physical address by use of translation tables (ST, PT) in a main storage (3) and the main storage (3) or a secondary storage (5) is accessed according to the translated physical address comprising:

first translation buffer means (40) including as entries therein pairs of upper-order portion (PAGETOP) of a plurality of instruction data existing in the main storage (3) and upper-order portions (LAU) of logical addresses (19) of said instruction data at addresses determined by intermediate portions (LAM) of the logical addresses;

second translation buffer means (41) including as entries therein pairs of upper-order portions (PAGETOP) of physical addresses of a plurality of operand data existing in the main storage (3) and upper-order portions (LAU) of logical addresses (19) of said operand data at addresses determined by intermediate portions (LAM) of the logical addresses;

select means (42, 43) responsive to a select signal (6) supplied from the processing unit (1) to indicate whether or not the memory access is directed to an instruction data or to an operand data, for selecting that one of entries read from the first and second translation buffer means (40, 41) at the time of memory access

which corresponds to the select signal (6); and

comparing means (44) for comparing the upper-order portion of a logical address in the entry selected by said select means (42, 43) with the upper-order portion of the logical address inputted and for outputting a signal (36) indicating that a physical address obtained by combining the upper-order portion in the selected entry of the physical address with the lower-order portion of the logical address inputted is valid.

2. An address translation circuit according to Claim 1 wherein said select means (42, 43) comprises:

first selector means (42) for receiving as inputs thereto the upper-order portion (LAU) of a logical address in the entry read from said first translation buffer means and the upper-order portion of a logical address in the entry read from said second translation buffer means and for supplying one of the upper-order portions to said select means in response to the select signal; and

second selector means (43) for receiving as inputs thereto the upper-order portion (PAGETOP) of a physical address in the entry read from said first translation buffer means and the upper-order portion (PAGETOP) of a physical address in the entry read from said second translation buffer and for outputting one of the upper-order portions in response to the select signal.

FIG. 1

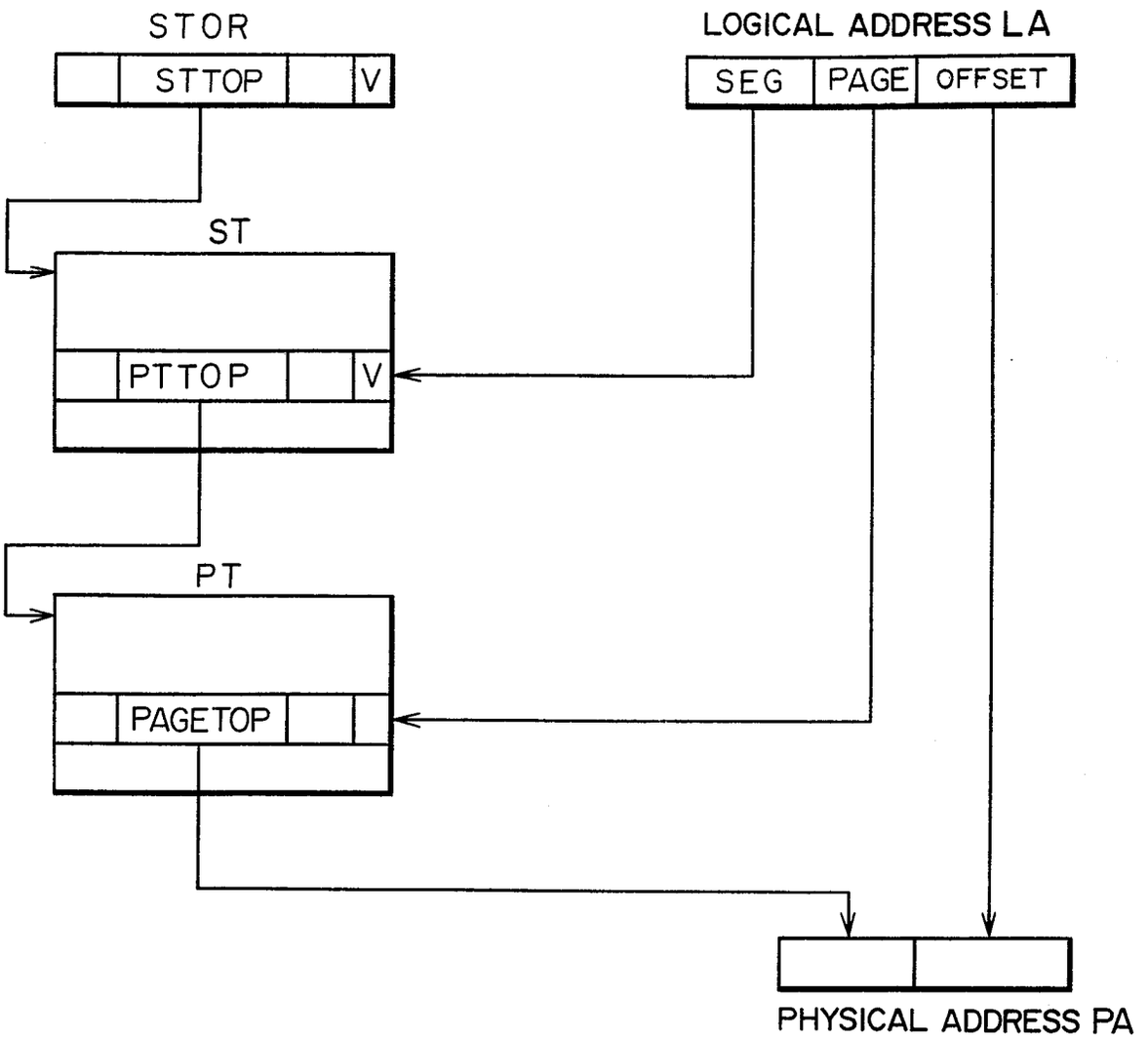

STOR
STTOP
V
LOGICAL ADDRESS LA
SEG
PAGE
OFFSET
ST
PTTOP
V
PT
PAGETOP
PHYSICAL ADDRESS PA

FIG. 2

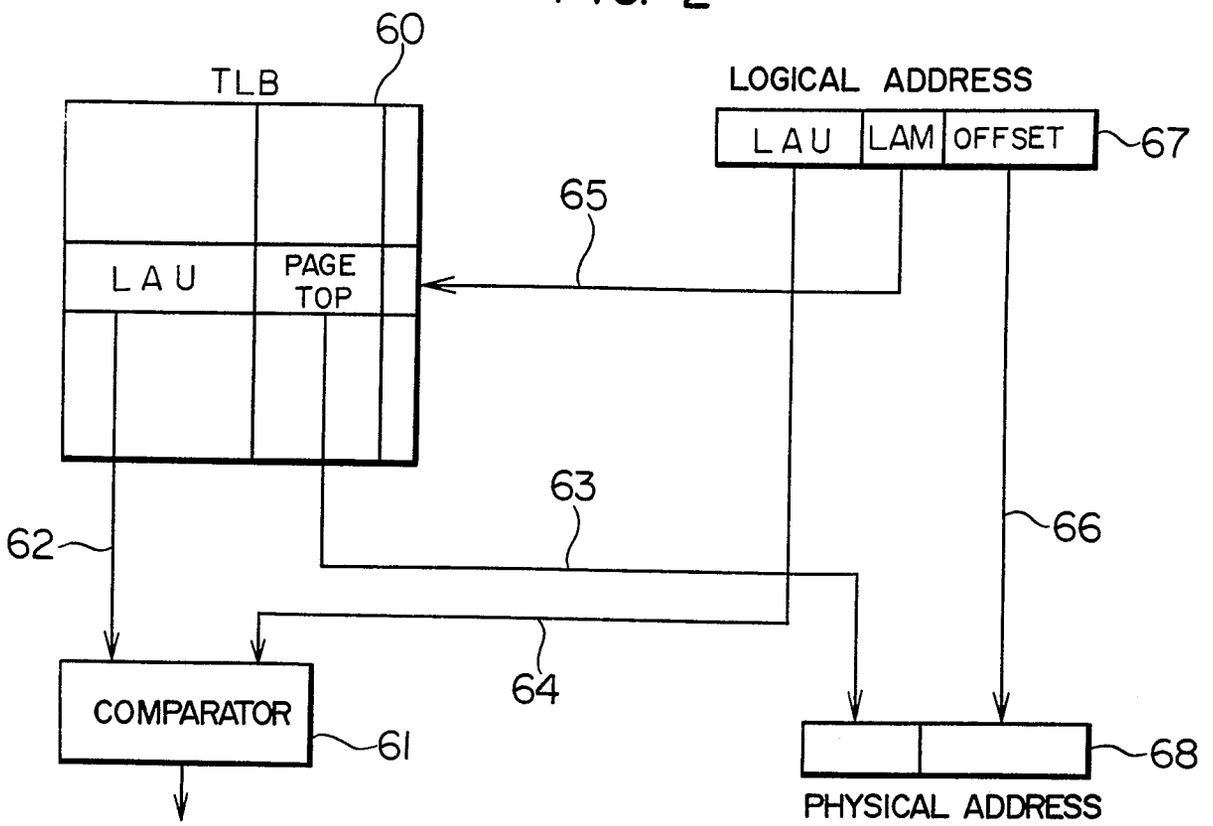
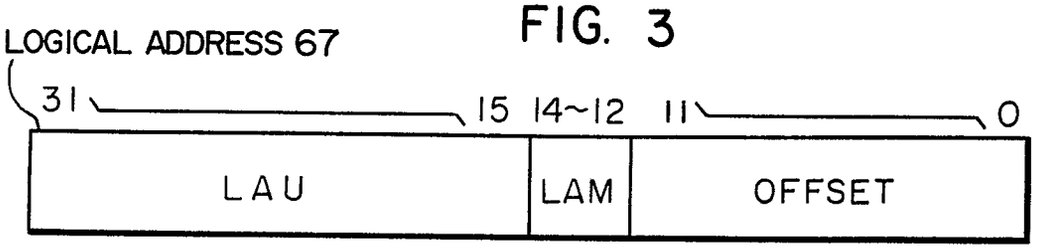
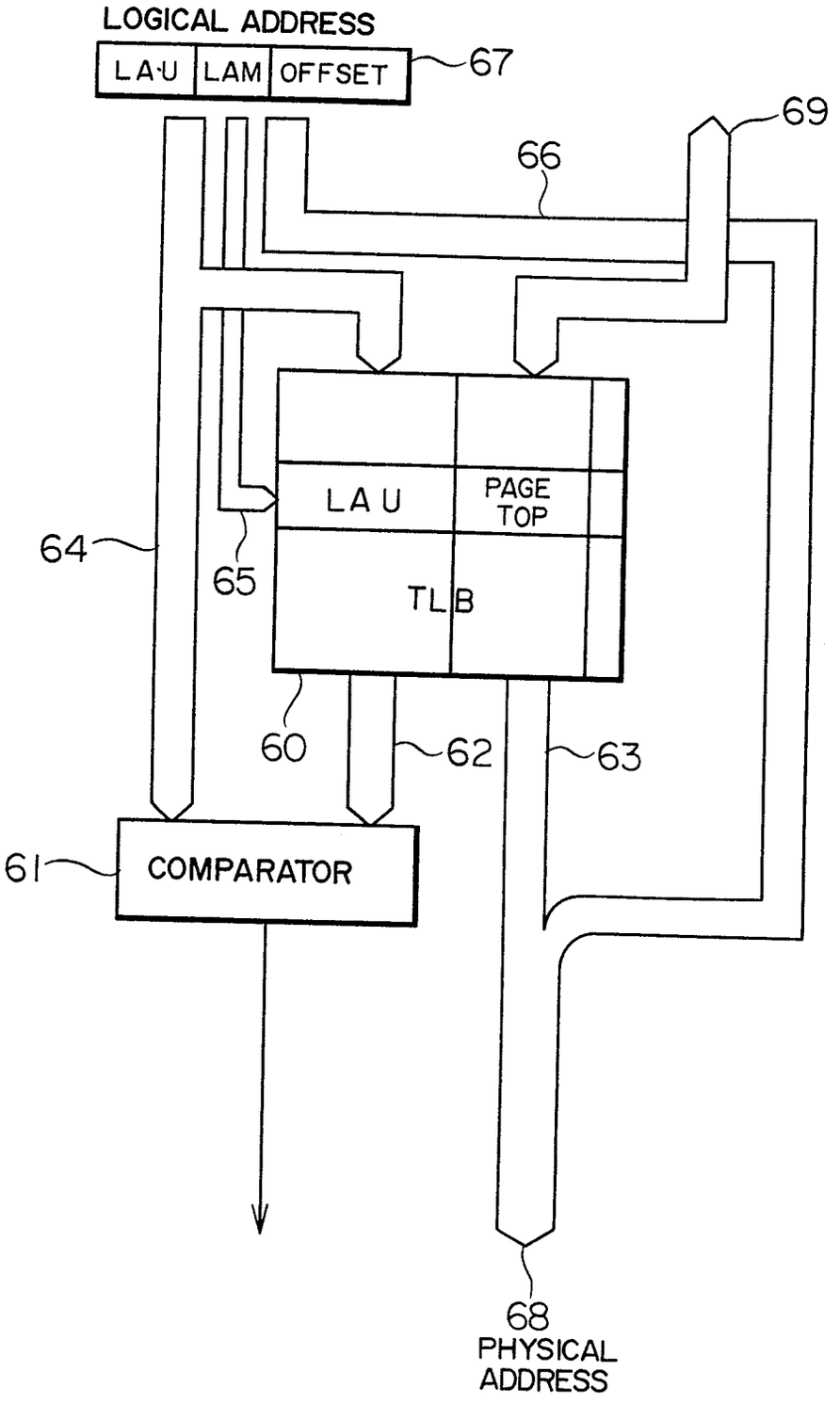


FIG. 3



LAM			TLB
14	13	12	ENTRY No.
0	0	0	0
0	0	1	1
0	1	0	2
0	1	1	3
1	0	0	4
1	0	1	5
1	1	0	6
1	1	1	7

FIG. 4


LOGICAL ADDRESS
LA U
LAM
OFFSET
67
69
66
LA U
PAGE TOP
TLB
64
65
60
62
63
61
COMPARATOR
68
PHYSICAL ADDRESS

4 / 9

FIG. 5

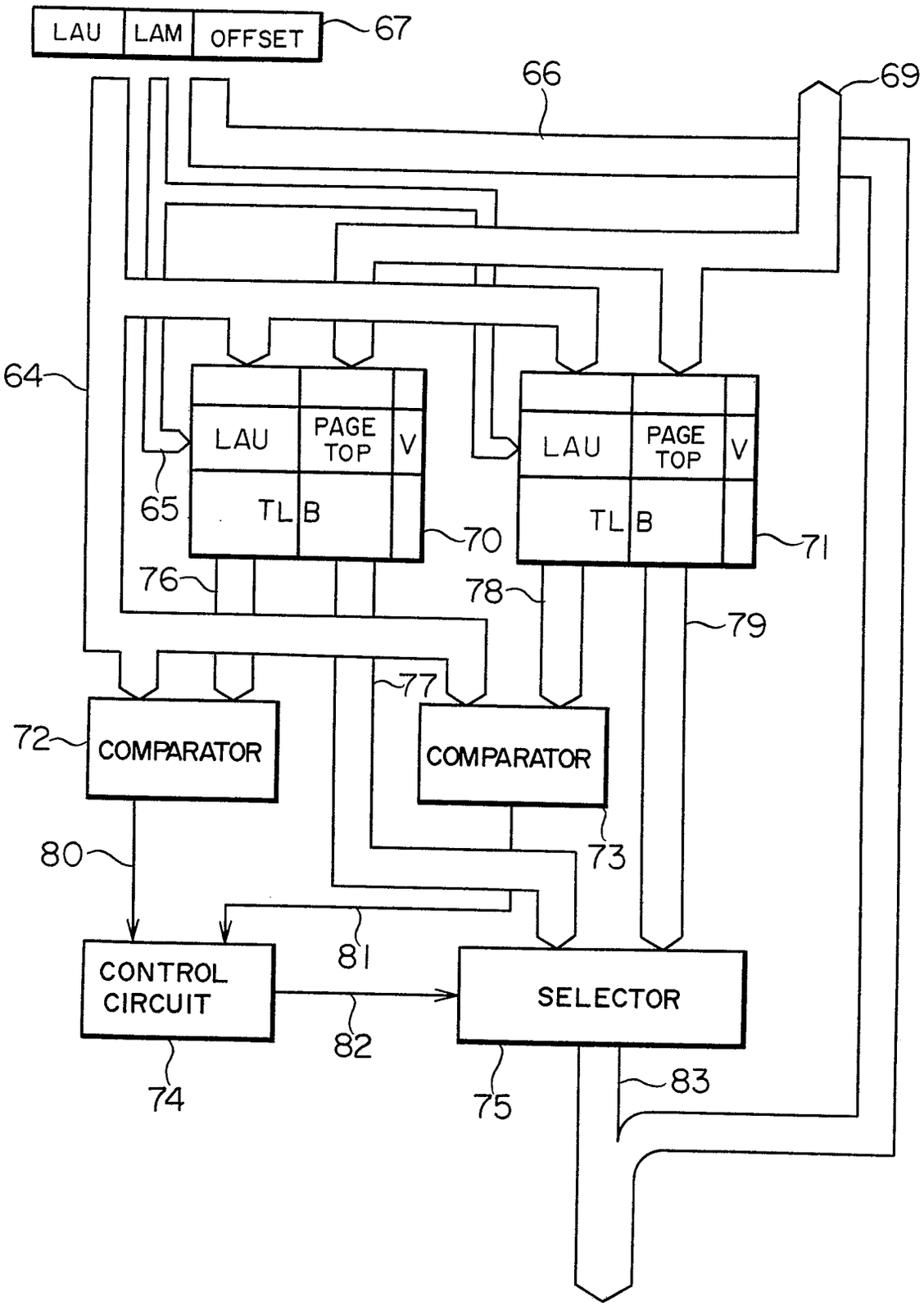

LAU
LAM
OFFSET
67
66
69
64
LAU
PAGE TOP
V
TLB
65
70
LAU
PAGE TOP
V
TLB
71
76
78
79
77
72
COMPARATOR
COMPARATOR
80
73
81
CONTROL CIRCUIT
SELECTOR
82
74
75
83

FIG. 6

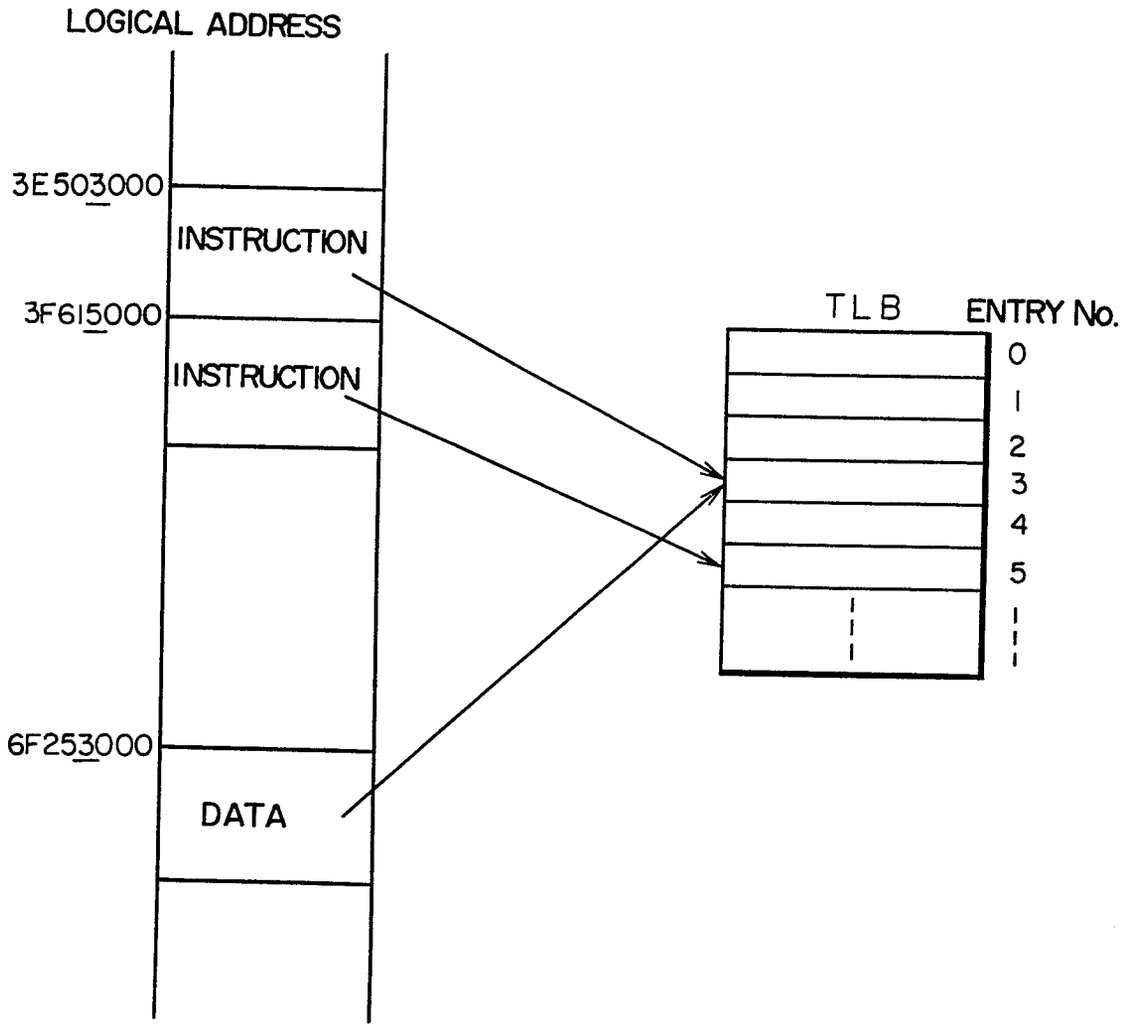

LOGICAL ADDRESS
3E503000
INSTRUCTION
3F615000
INSTRUCTION
6F253000
DATA
TLB
ENTRY No.
0
1
2
3
4
5

FIG. 7

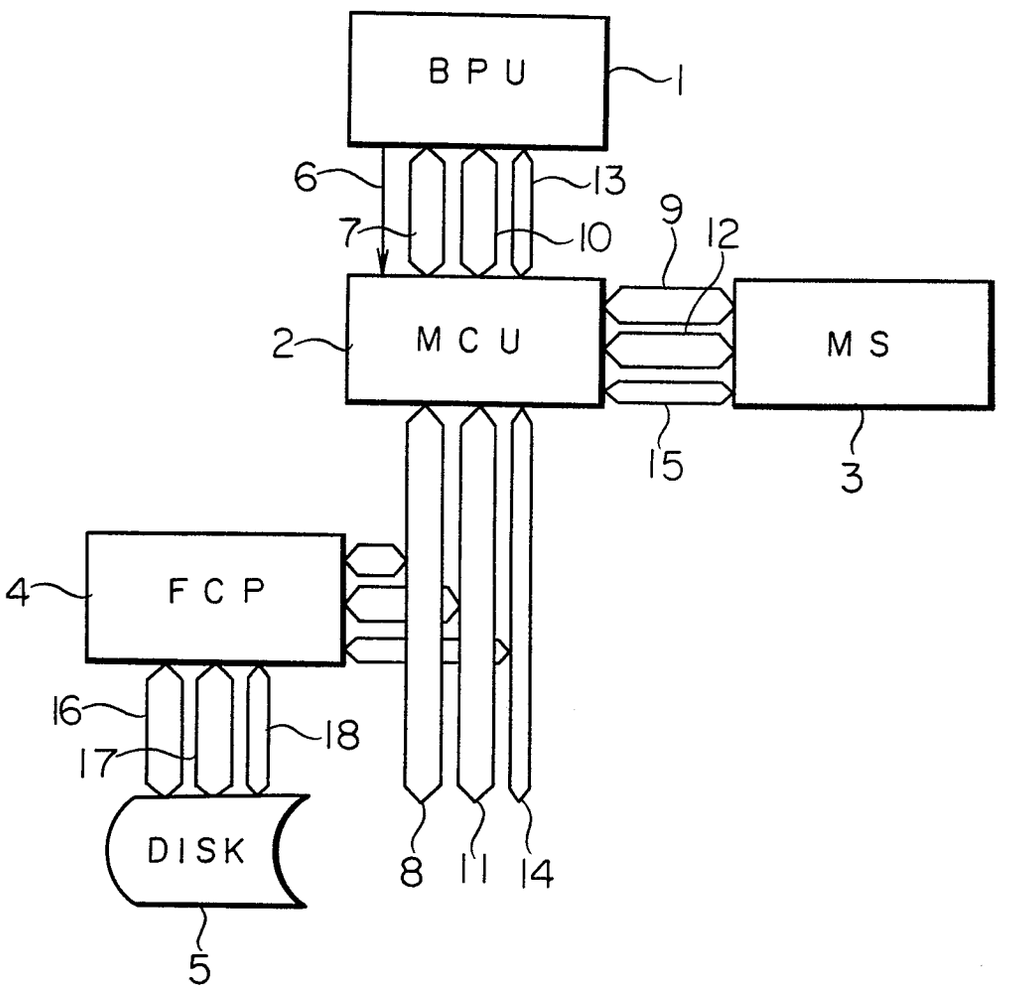


FIG. 8

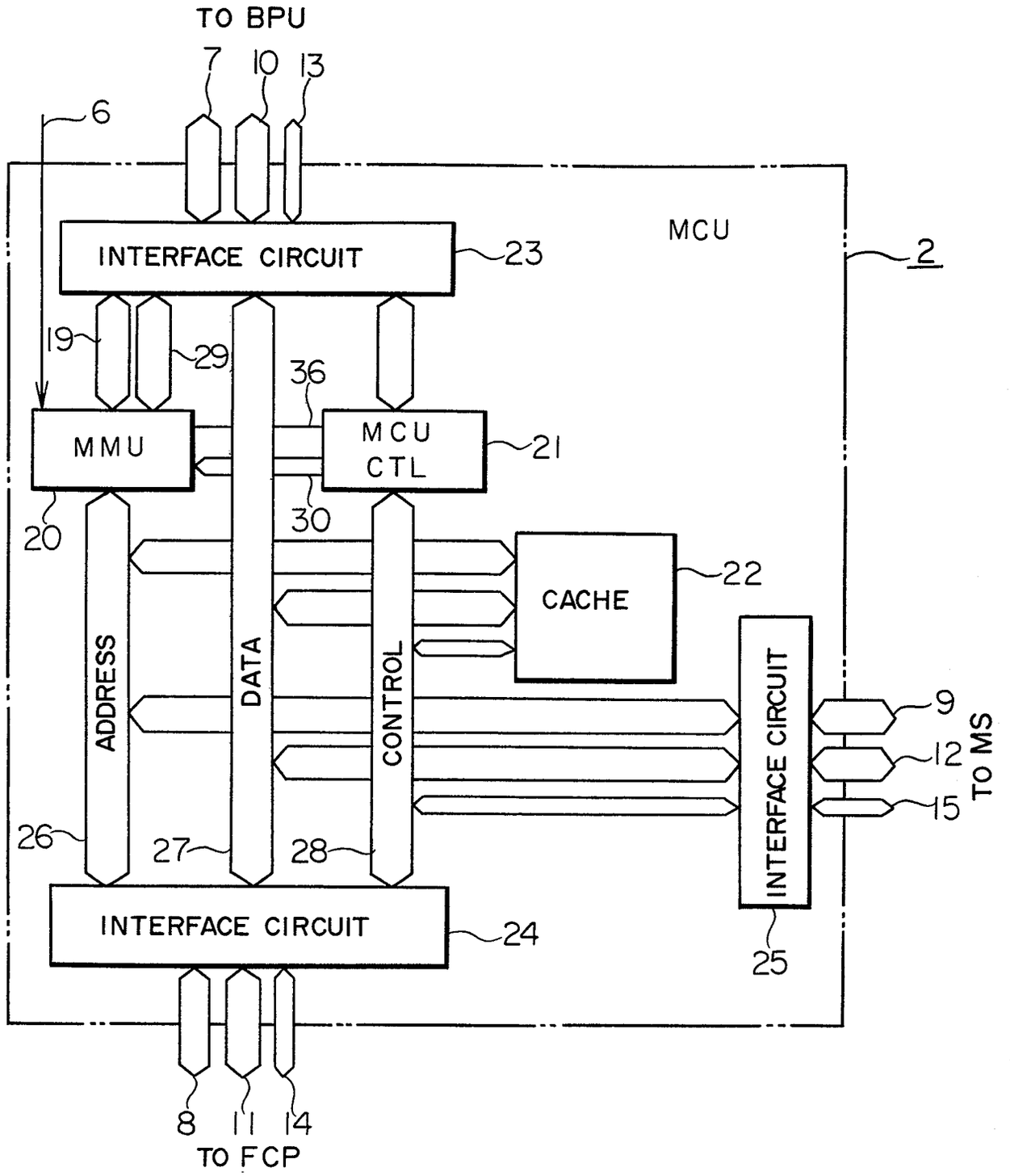

TO BPU
7
10
13
6
MCU
2
INTERFACE CIRCUIT
23
19
29
36
MMU
MCU CTL
21
20
30
CACHE
22
ADDRESS
DATA
CONTROL
INTERFACE CIRCUIT
9
12
15
TO MS
26
27
28
INTERFACE CIRCUIT
24
25
8
11
14
TO FCP

FIG. 9

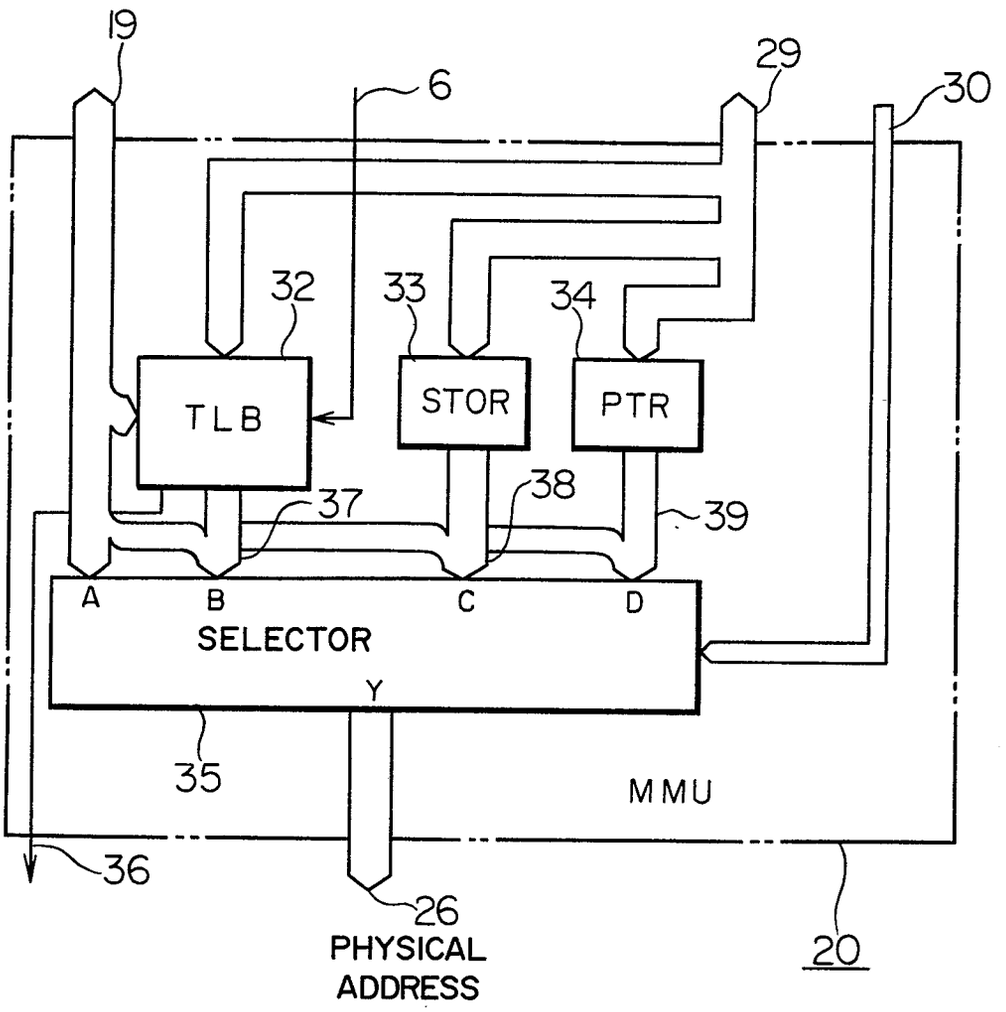
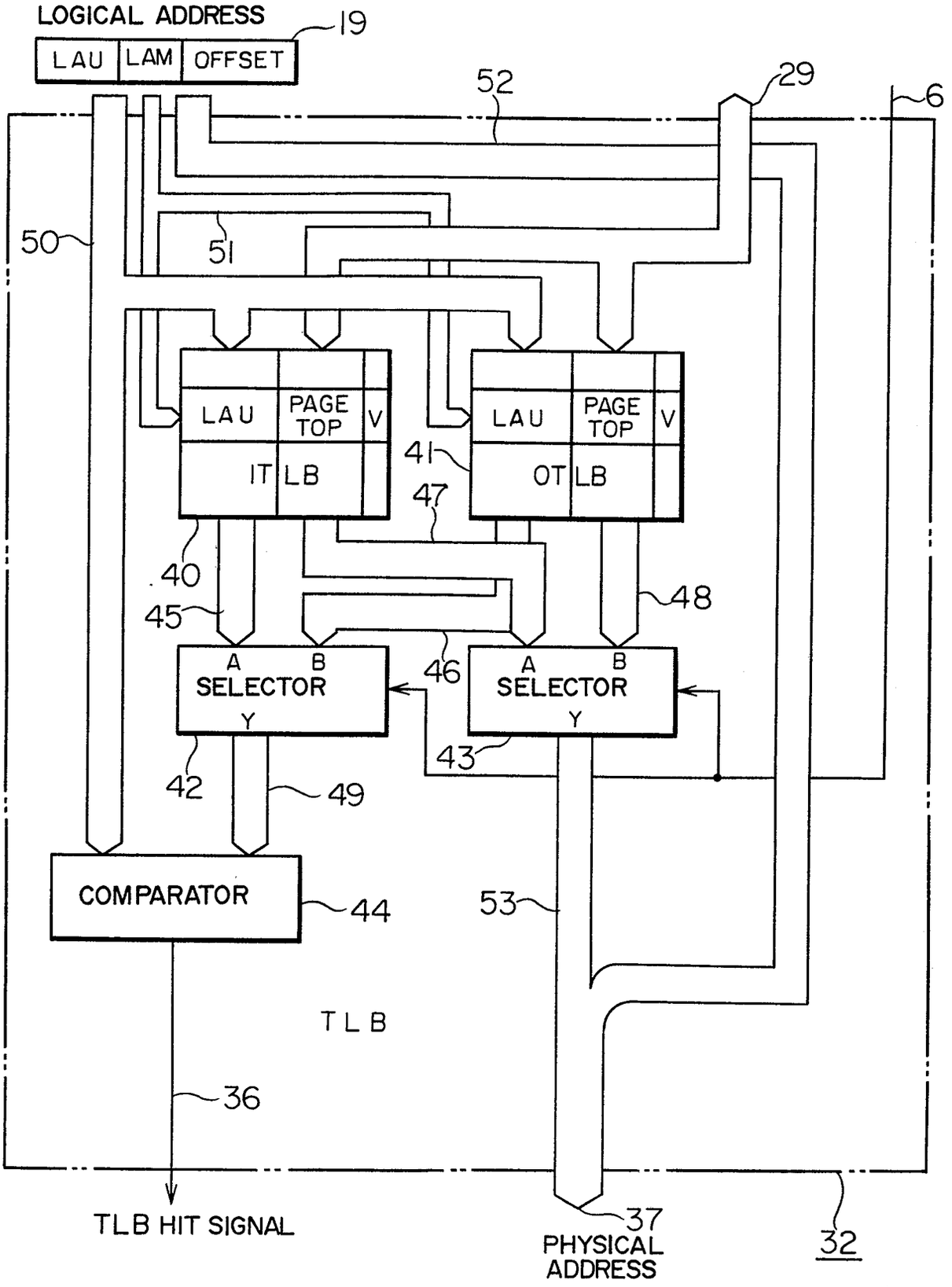

19
6
29
30
32
33
34
TLB
STOR
PTR
37
38
39
A
B
C
D
SELECTOR
Y
35
MMU
36
26
PHYSICAL
ADDRESS
20

FIG. 10


LOGICAL ADDRESS
19
LAU
LAM
OFFSET
52
29
6
50
51
LAU
PAGE TOP
V
IT
LB
41
OT
47
40
45
48
A
B
SELECTOR
Y
46
43
42
49
COMPARATOR
44
53
TLB
36
TLB HIT SIGNAL
37
PHYSICAL ADDRESS
32